

DIFFUSION SYNTHETIC ACCELERATION – PART II: KRYLOV METHODS  
FOR MULTI-DIMENSIONAL HETEROGENEOUS PROBLEMS

James. S. Warsa, Todd A. Wareing and Jim E. Morel  
Transport Methods Group  
Los Alamos National Laboratory  
Los Alamos, New Mexico 87545

Send proofs to:

James S. Warsa  
Los Alamos National Laboratory  
CCS-4, MS D409  
Los Alamos, NM 87545

Number of pages: 28  
Number of figures: 4  
Number of tables: 7

E-mail: *warsa@lanl.gov*  
FAX: 505-665-5538

# DIFFUSION SYNTHETIC ACCELERATION – PART II: KRYLOV METHODS FOR MULTI-DIMENSIONAL HETEROGENEOUS PROBLEMS

James S. Warsa, Todd A. Wareing and Jim E. Morel

Transport Methods Group

Los Alamos National Laboratory

Los Alamos, New Mexico 87545

## Abstract

In this paper we present a method to address the degradation of diffusion synthetic acceleration (DSA) methods in problems with material discontinuities and high scattering ratios, as shown in Part I. Our solution is to simply replace traditional source iteration with a Krylov iterative method that is preconditioned with the partially consistent simplified WLA DSA scheme. We describe how we can solve the lumped, linear discontinuous discretization of the first order form of the  $S_N$  transport equation on unstructured tetrahedral meshes using the Krylov iterative method GMRES. One attractive feature of this approach is that it can be implemented in terms of the original  $S_N$  space and angle sweeps with very little code modification. Computations on problems for which DSA-accelerated source iteration is impractical are compared to solutions with preconditioned GMRES. Results will show that the partially consistent DSA scheme is an effective preconditioner and such problems can now be solved efficiently.

## 1 INTRODUCTION

In the first part of this paper, we identified a class of problems that contained multiple highly diffusive materials with discontinuous total cross sections. In these types of problems, traditional DSA schemes, both fully and partially consistent, lost their effectiveness in accelerating the source iteration process. In Part I, we considered a lumped, linear discontinuous finite element (DFEM) discretization of the  $S_N$  transport equation on three-dimensional unstructured meshes.<sup>1,2</sup> We presented a Fourier analysis and numerical examples to illustrate the deficiencies of two DSA schemes for those kinds of problems. In this part, we present a way to solve this class of problems efficiently by simply replacing accelerated source iterations with a Krylov iterative method that is preconditioned with DSA.

Our approach follows the work of Ashby, et al.<sup>3</sup> and Guthrie, et al.,<sup>4</sup> where the Krylov iterative method GMRES, preconditioned by DSA, replaces traditional source iteration on the scalar flux in one-dimensional slab geometry  $S_N$  transport problems. We extend their approach to three-dimensional unstructured tetrahedral grids. The kinds of difficult problems presented in Part I converge unacceptably slowly because of a degradation in the spectral radius. Our implementation of the Krylov method on these kinds of problems shows that (a) the unpreconditioned Krylov method accelerates the transport solution, although not always as well as accelerated source iteration, (b) convergence is vastly improved if the Krylov method is preconditioned with DSA (c) the Krylov method restores the effectiveness of the simplified WLA (S-WLA) method, which is a partially consistent DSA scheme.<sup>5</sup> While this last observation is probably also true for the fully consistent DSA (FCDSA) method, the high cost associated with solving the fully consistent equations is unacceptable, particularly because it is shown in Part I that the degradation in its effectiveness is similar to that of the partially consistent method in the presence of material discontinuities. Therefore we will only consider the less costly S-WLA method for preconditioning the Krylov iterations. We will find that any extra computational overhead associated with the Krylov solver is outweighed by the drastic reduction in the number of iterations relative to S-WLA accelerated source iteration, although the relative improvement decreases with decreasing scattering ratio.

A nice feature of the Krylov method approach is that it can be “wrapped around” the original source iteration code so that only minor changes to the inner iteration code is necessary. Energy dependent iterations or other outer iterations would require more significant code modification. Extension to these types of problems is straightforward but comes at the cost of increased computer memory requirements. In this paper we consider only steady-state, one-group problems with isotropic scattering, but the method extends easily to anisotropic scattering. Extensions of the Krylov iterative method will be considered in future work.

The paper is organized as follows. We will discuss the implementation of the Krylov solution method in the next section. The following section will present numerical results using the AttilaV2 host code for some problems of the kind discussed in Part I. The paper concludes with some conclusions drawn from our

implementation and directions for further work.

## 2 KRYLOV ITERATIVE METHODS APPLIED TO TRANSPORT PROBLEMS

We will begin this section with a discussion of the relevant previous work on applying a Krylov iterative method to first order,  $S_N$  transport. Then we will present what we will refer to as the scalar flux formulation of the  $S_N$  transport equation. The section closes with a discussion of some of the implementation issues that had to be addressed.

### 2.1 Related Work

There has been a fair amount of work during the past decade in applying Krylov methods to transport problems in various contexts. The paper by Faber and Manteuffel<sup>6</sup> analyzed the properties of DSA and preconditioned conjugate gradients (PCG) to solve symmetric forms of the transport equation in one dimension. Kelley<sup>7</sup> used GMRES at the lowest level of a multi-level acceleration scheme and Kelley and Xue<sup>8</sup> then considered the use of GMRES in the context of accelerating solutions of the transport equation and how properties of integral operators can be used to improve the GMRES iterations in one dimension. Oliveira and Deng considered nonsymmetric Krylov methods<sup>9</sup> and iterated on the angular flux in one dimension. They showed results for GMRES, CGS and CGNE Krylov methods (see Ref. 10 for a description and definition of these and other related methods) preconditioned with ILU, spatial multigrid and angular multigrid techniques. Patton and Holloway<sup>11</sup> also considered the nonsymmetric equations for the angular flux and compared GMRES with and without DSA preconditioning to source iterations in one dimension. Sanchez and Santandrea<sup>12</sup> consider a symmetrization of the transport operator and use the Lanczos algorithm (which is mathematically equivalent to CG if the operator is positive definite) with the method of characteristics in two dimension and compare the results to DSA. Patton and Holloway<sup>13</sup> investigated using preconditioned GMRES for the angular flux, including energy dependence and algebraic preconditioners such as incomplete LU factorizations in one dimension.

Ashby, et al.<sup>3,14</sup> and Brown,<sup>15</sup> and later Guthrie, et al.<sup>4</sup> formulated the problem in terms of the scalar flux (moments). This has the advantage of significantly reducing the dimension of the problem. It is interesting, however, to note that Guthrie, et al., did not reference these earlier papers. Greenbaum<sup>16</sup> also used this approach in her textbook where she considered one dimensional slab geometry transport preconditioned with DSA to illustrate how nonsymmetric Krylov solvers can be used for solving “real” problems. Ramone, et al.,<sup>17</sup> used such an approach in one dimensional slab geometry to solve the low-order transport synthetic acceleration equations with PCG. Zika and Adams<sup>18</sup> extended this approach two two-dimensional Cartesian geometry.

The ideas most influential on the work we are presenting here those of Ashby, et al.,<sup>3,14</sup> and Guthrie, et al.<sup>4</sup> Ref. 3 appears to be the first work exploring the merits of Krylov subspace methods (GMRES) relative

to the Richardson (source) iteration that has traditionally been used in transport applications in the general nonsymmetric case. Both diamond and linear discontinuous differencing of the  $S_N$  equations were considered. This paper was apparently a preliminary version of the very detailed linear algebraic analysis, also in one dimension, of DSA and transport solution methods in Ref. 14. This was followed by the extension of their analysis to three-dimensional cartesian geometry in Ref. 15. The last two papers worked with the diamond differenced  $S_N$  equations, and computed the spectral radius of the preconditioned linear system explicitly for a variety of physical situations, including the thick and thin asymptotic limits. They analyzed the effectiveness and efficiency of using both consistent and inconsistent DSA methods for source iteration, Chebyshev iteration, and GMRES. Ref. 4 provides an enlightening discussion on how GMRES generates an optimal linear combination of the previous scalar flux iterates. They present their approach as a multi-step acceleration scheme, even without DSA as a preconditioner to GMRES. Their numerical results address not only the effectiveness of the approach in improving convergence but also the computational efficiency relative to accelerated source iterations.

These papers show similar numerical results with similar conclusions regarding the effectiveness and efficiency of GMRES preconditioned with DSA. However, Ashby, et al.,<sup>14</sup> and Brown,<sup>15</sup> put emphasis on the analysis of DSA as a preconditioner and the linear algebraic formalism that facilitates their analysis while Guthrie, et al.<sup>4</sup> focused primarily on how to implement the iterative solution in terms of transport sweeps and the properties of the Krylov method iterative solution. In our view, the most important feature of all these works is that the matrix-vector operations that need to be computed in the course of a Krylov iterative solution can be carried out without explicitly forming a matrix. It can be implemented via the usual space-angle transport sweep, scattering integral calculations, and scalar flux moment computations. Considering other work on Krylov iterative methods on the full angular flux we feel that the scalar flux formulation they presented is more promising, especially if the method is implemented within an outer iteration. This is because the Krylov iterative method can be implemented with only minor code modification and a reasonable amount of extra storage. As explained in Ref. 4, the Krylov method combines previous iterates in such a way as to accelerate convergence. As long as the additional computational overhead of the Krylov method is not too great then it seems reasonable to “augment” the usual source iteration process and make use of information that has already been computed.

Ashby, et al.,<sup>3</sup> considered an inconsistent DSA discretization whose effectiveness degraded in the presence of two very different materials. While it was totally ineffective in accelerating source iteration they found that convergence improved significantly when it was applied as a preconditioner to GMRES. Their work presaged the difficulties we encountered in Part I of this paper. They state in their conclusion, “These results have possible implications for problems in higher spatial dimensions, for which a consistent preconditioner is difficult to obtain and/or impractical to apply.” Furthermore, as Brown<sup>15</sup> points out, if the diffusion

equation in the DSA algorithm isn't solved exactly or some other approximations are made, "...the use of these methods in three dimensions will be crucial to the overall usefulness of DSA in 3-D problems," referring to "...more powerful iterative methods such as Bi-CGSTAB ...". Indeed, this is exactly the case for our linear discontinuous  $S_N$  transport discretization on three-dimensional unstructured tetrahedral grids. The FCDSA algorithm is prohibitively expensive to employ. Only the partially consistent S-WLA DSA method remains feasible for general purpose. This is because, as a preconditioner to a Krylov iterative method, it remains effective even in the kinds of heterogeneous problems we are considering here. Furthermore, the overall transport solution can be computed efficiently because the S-WLA method computations are inexpensive.

## 2.2 Scalar Flux Formulation

In this section we will present the scalar flux formulation that we use in solving the transport equation. We will work in a general operator notation. Details of the definitions of these operators can be found in Part I of this paper. Vectors are lower case having dimension  $n = N(L + 1)$  where  $N$  is the number of spatial grid unknowns and  $L$  is the order of scattering anisotropy. For the linear discontinuous discretization,  $N = 4N_{cells}$ , where  $N_{cells}$  is the number of tetrahedral cells in the mesh. Operators are of order  $(n \times n)$  denoted in bold uppercase;  $\mathbf{I}$  represents the identity.

Source iteration, or Richardson iteration, is probably the most commonly used iterative solution method for the first order  $S_N$  transport equation because it can be efficiently implemented via an angular sweep.<sup>1</sup> In standard notation the discrete ordinates transport equation is

$$\mathbf{L}_m \psi_m = \mathbf{S} \mathbf{K} \psi_m + Q_0, \quad \mathbf{B} \psi_m = g_m, \quad (1)$$

and  $\psi_m$  is the angular flux in quadrature direction  $m$ ,  $\mathbf{L}_m$  is the streaming-plus-removal operator for angle  $m$  which is inverted by the angular sweep algorithm,  $\mathbf{S}$  represents the scattering operator,  $\mathbf{K}$  is the angular-flux-to-moment operator, and  $Q_0$  is an isotropic source of particles. The boundary conditions on the angular fluxes,  $\mathbf{B} \psi_m$ , are a function of angle depending on incoming and outgoing directions relative to the orientation of the quadrature direction with respect to the boundary faces. We indicate this dependence by the angular function  $g_m$ . We consider only the standard vacuum, source, and reflective conditions. Ignoring boundary conditions for the time being, in terms of the scalar flux moments, Eq. 1 is solved with the iteration

$$\phi^{\ell+1} = \mathbf{T} \mathbf{S} \phi^\ell + b, \quad (2)$$

given some initial guess  $\phi^0$  to the scalar flux. Here the operator  $\mathbf{T} = \mathbf{K} \mathbf{L}_m^{-1}$  and  $b$  is a source vector. Again, more details can be found in Part I.

With DSA, the iteration is modified by a correction step as follows

$$\phi^{\ell+1/2} = \mathbf{T}\mathbf{S}\phi^\ell + b \quad (3a)$$

$$f^{\ell+1/2} = \mathbf{D}^{-1}\mathbf{S}(\phi^{\ell+1/2} - \phi^\ell) \quad (3b)$$

$$\phi^{\ell+1} = \phi^{\ell+1/2} + f^{\ell+1/2}, \quad (3c)$$

The operator  $\mathbf{D}^{-1}$  involves the inverse of the diffusion equation or the inverse of the  $P_1$  equations. It may consist of several other operations as well. In the case of the S-WLA algorithm, for example, involves the approximate inversion of the vertex-centered, linear continuous finite element discretization of the diffusion equation. The solution to that SPD linear system can be computed efficiently using preconditioned conjugate gradients (PCG). The source term for the solution is calculated by projecting the discontinuous scalar flux residual  $\mathbf{S}(\phi^{\ell+1/2} - \phi^\ell)$  onto the vertices. The solution is then interpolated to compute a correction to the discontinuous zeroth-order transport scalar flux moments.

We can show that the DSA algorithm is equivalent to a preconditioning the transport operator as follows. First, consider that Richardson iteration for some linear system  $\mathbf{A}x = y$  is simply

$$\begin{aligned} x^{\ell+1} &= x^\ell + r^\ell \\ &= x^\ell + (b - \mathbf{A}x^\ell) \\ &= (\mathbf{I} - \mathbf{A})x^\ell + y. \end{aligned} \quad (4)$$

Comparing this with Eq. 2, we find that the linear operator corresponding to source iteration is  $\mathbf{A} = (\mathbf{I} - \mathbf{T}\mathbf{S})$ . Now Richardson iteration for the (left) preconditioned linear system  $\mathbf{M}^{-1}\mathbf{A}x = \mathbf{M}^{-1}y$  is

$$\begin{aligned} x^{\ell+1} &= x^\ell + \mathbf{M}^{-1}r^\ell \\ &= x^\ell + \mathbf{M}^{-1}(y - \mathbf{A}x^\ell) \\ &= (\mathbf{I} - \mathbf{M}^{-1}\mathbf{A})x^\ell + \mathbf{M}^{-1}y. \end{aligned} \quad (5)$$

Recall that preconditioning will be effective if  $\mathbf{M}^{-1}$  is in some sense an approximation to  $\mathbf{A}^{-1}$ . The preconditioner may be computed explicitly in advance or it may involve the solution of another linear system,  $\mathbf{M}w = z$ , for example, which might need to be computed iteratively or approximately at every iteration. The overall solution will be computed more efficiently only if the preconditioning system can be computed relatively easily.

Collapsing the DSA algorithm, Eqs. 3, into a single operation gives

$$\begin{aligned}\phi^{\ell+1} &= \mathbf{TS}\phi^\ell + b + \mathbf{D}^{-1}\mathbf{S}(\mathbf{TS}\phi^\ell + b - \phi^\ell) \\ &= [\mathbf{I} + (\mathbf{I} + \mathbf{D}^{-1}\mathbf{S})(\mathbf{TS} - \mathbf{I})]\phi^\ell + (\mathbf{I} + \mathbf{D}^{-1}\mathbf{S})b\end{aligned}\tag{6}$$

which, by comparison with the Eq. 5, shows that the DSA algorithm is just Richardson iteration for the preconditioned system

$$(\mathbf{I} + \mathbf{D}^{-1}\mathbf{S})(\mathbf{I} - \mathbf{TS})\phi = (\mathbf{I} + \mathbf{D}^{-1}\mathbf{S})b.\tag{7}$$

We can see that  $(\mathbf{I} + \mathbf{D}^{-1}\mathbf{S})$  is in fact an approximation to the inverse of  $(\mathbf{I} - \mathbf{TS})$  as follows. The computation of the error estimate in Eqs. 3,  $f^{\ell+1/2} = \mathbf{D}^{-1}\mathbf{S}(\phi^{\ell+1/2} - \phi^\ell)$ , is an approximation to the error equation

$$(\mathbf{I} - \mathbf{TS})f^{\ell+1/2} = \mathbf{TS}r^\ell,\tag{8}$$

where  $f^{\ell+1/2} = (\phi - \phi^{\ell+1/2})$ ,  $r^\ell = (\phi^{\ell+1/2} - \phi^\ell)$ , and  $\phi$  is the exact solution to Eq. 2. Examining Eqs. 3 it is evident that

$$\mathbf{D}^{-1}\mathbf{S} \approx (\mathbf{I} - \mathbf{TS})^{-1}\mathbf{TS} = (\mathbf{I} - \mathbf{TS})^{-1} - \mathbf{I},\tag{9}$$

or, in other words,

$$(\mathbf{I} + \mathbf{D}^{-1}\mathbf{S}) \approx (\mathbf{I} - \mathbf{TS})^{-1}.\tag{10}$$

It is the preconditioned linear system, Eq. 7, that we will solve with a Krylov iterative method, which we implemented as follows. Assume for the moment that we are solving the preconditioned linear system  $\mathbf{M}^{-1}\mathbf{A}x = \mathbf{M}^{-1}y$ . At every iteration, the Krylov method supplies a vector  $v$  to which the linear system is applied, that is, the vector  $z = \mathbf{A}v$  is computed and returned to the Krylov solver. Subsequently, the linear system  $\mathbf{M}w = z$  is “solved” and the vector  $w$  is returned to the solver. This allows us to compute the “action” of the preconditioner on the vector  $z$  without the inverse matrix  $\mathbf{M}^{-1}$  being available.

However, we do not actually use the preconditioned version of the Krylov algorithm where we would have to first apply the linear system to  $v$ ,  $z = (\mathbf{I} - \mathbf{TS})v$ , and subsequently compute the action of the preconditioner on  $z$ ,  $w = (\mathbf{I} + \mathbf{D}^{-1}\mathbf{S})z$ , as we would if DSA were being used as a preconditioner. Instead, we use the unpreconditioned version of the Krylov method. At every iteration we set  $\phi^\ell = v$ , compute the sequence of operations shown in the DSA algorithm, Eqs. 3, with the original source iteration code. We then return  $w = v - \phi^{\ell+1}$  to the Krylov solver, which completes the operator  $(\mathbf{I} - \mathbf{TS})$ . Collapsing the solution process into a single operation shows that this approach is fully equivalent to preconditioning in the usual sense but requires much less code modification because the Krylov solver can just be “wrapped around” the source iteration code with the addition of just the final step  $w = v - \phi^{\ell+1}$ . Implementing the iteration this way makes it much easier to treat reflective boundary conditions, which will be discussed in greater detail

below.

## 2.3 Implementation Issues

There are several issues which had to be addressed in the course of implementing the Krylov iterative solution method. We will discuss a few of them now, but before doing so it should be noted that we have tried to make the implementation of the Krylov iterative solver approach as efficient as possible but there is still room for improvement. AttilaV2 is currently a serial program written in Fortran 90 so our approach and conclusions might change if the method is implemented in parallel.

### 2.3.1 Choice of iterative method.

There are many choices to be made when choosing a Krylov iterative solution method. One typically chooses an iterative solver when a matrix is sparse, when direct solution methods are impractical, or when the matrix is not explicitly computed. In our case we do not explicitly compute the discrete  $S_N$  transport equation matrix because source iteration is implemented via an angular sweeping algorithm which is matrix-free and very efficient.

We will review how we arrived at our choice of using GMRES. The reader can consult any number of recent texts on Krylov methods for further information including, for example, Refs. 10, 16, or 19. The first choice concerns the symmetry of the matrix. The CG, MINRES and SYMMLQ methods are used for symmetric systems; the first of these is used if the linear system is positive definite and the latter two if it is indefinite. These methods are very efficient and require additional storage consisting of just a few auxiliary vectors. The transport operator, as formulated in Eq. 1, is symmetric only under certain circumstances. We have therefore decided to use a method intended for nonsymmetric linear systems in order to make a fair comparison with source iteration which is not constrained by symmetry considerations. Also, further possible complications such as symmetrization of the transport operator<sup>12</sup> or the need for specialized quadratures or inner products, will not be necessary, so we can implement the Krylov iterations with very little change to the original source iteration algorithm. We must therefore use a method intended for nonsymmetric linear systems. These include only the transpose-free methods, such as GMRES (Generalized Minimal Residual), TFQMR (Transpose-Free Quasi-Minimal Residual), Bi-CGSTAB (Bi-Conjugate Gradient Stabilized), or CGS (Conjugate Gradient Squared), because we cannot easily use methods for which the transpose of the linear system is required.

In our experience we have found GMRES to be the most reliable and most rapidly converging nonsymmetric Krylov method.<sup>2,20</sup> The convergence behavior of simple iterative methods like Richardson iteration or Krylov iterative methods for symmetric systems is well-understood in terms of the eigenvalues of the linear system. In contrast, there is very little that can be done to accurately predict or estimate the convergence of nonsymmetric Krylov iterative methods, except under special circumstances. Even if available

the estimated bounds may not be sharp. While this is not entirely satisfactory, we do not consider this lack of a priori knowledge about convergence to be restrictive because our computational experience so far indicates that GMRES is very robust for this application. Another possible drawback of GMRES is that storage and computational effort both increase linearly with every iteration. This can be limited to some extent by restarting every  $m$  iterations; the restarted version of GMRES is called GMRES( $m$ ). For reasons outlined below we have chosen the flexible version of GMRES( $m$ ), denoted FGMRES( $m$ ).

### 2.3.2 Distributed sources and boundary conditions.

The two concerns to be discussed now are the implementation of distributed sources and boundary conditions. We consider only isotropic sources, both boundary and distributed, in this paper.

In our implementation, the source vector  $b$  in Eq. 2 not only represents the uncollided flux due to distributed sources, as in Part I, but it also contains any incident boundary sources as well. This can be made clear by the following considerations (see also Ref. 3). Let  $\psi_m^0$  be the homogeneous solution to Eq. 1 with inhomogeneous boundary conditions:

$$\mathbf{L}_m \psi_m^0 = 0, \quad \mathbf{B} \psi_m^0 = g_m. \quad (11)$$

Now, given some right hand side,  $s$ , and homogeneous boundary conditions, the solution  $\psi_m$  to Eq 1 can be written as

$$\psi_m = \mathbf{L}^{-1} s, \quad \mathbf{B} \psi_m = 0. \quad (12)$$

Linearity implies that the complete solution to Eq. 1 with right hand side  $s$  and boundary condition  $\mathbf{B} \psi_m^0 = g_m$  is

$$\psi_m = \mathbf{L}^{-1} s + \psi_m^0. \quad (13)$$

Substituting the full right hand side, including the scattering operator, of Eq. 1 for  $s$  in this last expression gives

$$\psi_m = \mathbf{L}^{-1} (\mathbf{S} \mathbf{K} \psi_m + Q_0) + \psi_m^0. \quad (14)$$

Operating on this expression with angular flux-to-moment operator  $\mathbf{K}$  we find that

$$\phi = \mathbf{K} \mathbf{L}^{-1} \mathbf{S} \phi + \mathbf{K} (\mathbf{L}^{-1} Q_0 + \psi_m^0), \quad (15)$$

which is just Eq. 2 (iteration indices omitted) with the source term  $b = \mathbf{K} (\mathbf{L}^{-1} Q_0 + \psi_m^0)$ . Thus, we initialize the source vector  $b$  with a single transport sweep before we enter the Krylov iterative solver and then set any distributed or boundary sources to zero in subsequent calls to the transport sweep algorithm once the Krylov iterations begin.

In the host code, AttilaV2, specular reflection boundary conditions are constrained to be applied only on boundary faces which are oriented parallel to the coordinate axes. Reflecting boundary conditions are treated implicitly. This means that at any given iteration the incoming angular fluxes are assumed to not necessarily be equal to the outgoing angular fluxes on the reflective faces. Implicit treatment allows for the possibility of opposing reflecting boundary faces. It is not necessary if the transport sweep can be ordered such that the angular sweeps are started on reflecting boundary faces only after the sweep has computed all the outgoing angular fluxes on those faces. In AttilaV2, all reflecting faces are treated implicitly to simplify code logic, regardless of the transport sweep ordering, at the expense of additional storage. The angular flux unknowns on reflecting faces must be appended to Krylov iteration vector and iterated upon during the Krylov iterations because these angular fluxes, like the scalar fluxes, are fundamental unknowns in the problems. This is necessary whether reflection is treated implicitly or not.

Because the Krylov iteration is wrapped around the original source iteration coding, the only requirement needed to treat reflective conditions is to augment the vector of scalar flux moments with the angular fluxes on the reflective boundary. Thus, the transport iteration now has the form

$$\tilde{\phi}^{\ell+1/2} = \mathbf{T}\tilde{\mathbf{S}}\tilde{\phi}^{\ell} + \tilde{\mathbf{b}} \quad (16)$$

where the vector  $\tilde{\phi}$  contains not only the scalar flux moments  $\phi$  but also the boundary angular fluxes, represented by  $\psi_R$ , that is

$$\tilde{\phi} = \begin{bmatrix} \phi \\ \psi_{R,m} \end{bmatrix}, \quad (17)$$

and the source vector is now  $\tilde{\mathbf{b}} = [b \ 0]^T$ . The scattering operator is also augmented with the portion of the boundary condition operator that not only specifies vacuum and source conditions but also relates the incoming to outgoing angular fluxes on reflective faces. We can decompose the boundary operator into two parts, one for vacuum and source conditions,  $\mathbf{B}_0$ , and the other for reflective conditions,  $\mathbf{B}_R$ ,

$$\mathbf{B}\psi_m = \begin{bmatrix} \mathbf{B}_0 & \mathbf{B}_R \end{bmatrix} \begin{bmatrix} \psi_{0,m} \\ \psi_{R,m} \end{bmatrix}. \quad (18)$$

where the angular fluxes  $\psi_{0,m}$  are those on the boundary faces of the mesh with source or vacuum conditions and  $\psi_{R,m}$  are those on reflective boundary faces. The scattering operator is then augmented by the reflection operator as

$$\tilde{\mathbf{S}}\tilde{\phi} = \begin{bmatrix} \mathbf{S} & \mathbf{B}_R \end{bmatrix} \begin{bmatrix} \phi \\ \psi_{R,m} \end{bmatrix}. \quad (19)$$

Similarly, the error estimate calculation in the DSA algorithm also is modified to account for reflection.

In that case, the scattering operator also includes a projection of the residual in the scalar fluxes composed from the angular fluxes on the reflective boundary faces. The angular fluxes are subsequently corrected based on the diffusion equation solution. See Ref. 1 for more details. This discussion of the reflective conditions is presented to complete our abstract operator notation and to indicate why and how the scalar flux vector is augmented with the reflective boundary angular fluxes. Beyond that, no special treatment of reflective boundary conditions is necessary if the Krylov method is “wrapped around” the original source iteration algorithm as we described before.

### 2.3.3 Inner–outer iterations.

The DSA preconditioning of the Krylov iterative solution involves an inner conjugate gradient iteration for every outer iteration. The question of how to best implement combined inner–outer Krylov method iterations is an area of recent interest. There are two main considerations. The first is how the preconditioned Krylov subspace is constructed when the preconditioner can change or some other approximation is made that varies from iteration to iteration. This may be the case, for instance, when the preconditioner involves solving another linear system with an inner iteration computed to some tolerance, as in our case. This is addressed by saving the preconditioned vectors from iteration to iteration as implemented the flexible variants of Krylov methods that have been recently developed.<sup>21,22</sup> The second consideration is how to choose the inner iteration tolerance when using a nested pair of inner–outer Krylov methods so that the fewest inner iterations may be taken without affecting the outer iterative convergence too much in order to achieve the best possible overall computational efficiency. This has been addressed recently in Ref. 23 for solutions in which both the inner and outer method are CG. GMRES( $m$ ) is considered in Ref. 24. There, an inner iteration is simulated by calculating the matrix–vector multiplies approximately by perturbing (structure and size of) the original matrix at every GMRES iteration. Remarkably, they found that the perturbations could grow significantly during the iterations as long as the initial Krylov vectors are computed to high precision. That is, early in the outer iterative process the inner iterative solution needs to be computed to a strict convergence tolerance which can then be relaxed as the outer iteration proceeds. We implemented this idea in a way similar to that suggested in Ref. 24. We set the inner tolerance in inverse proportion to the norm of the outer residual vector (or its iteratively computed approximation). At restart, however, the inner solution should again be computed with high precision and the tolerance relaxed subsequently.

We have used both of these approaches in our solution method. In the results which follow, the inner tolerance for the DSA conjugate gradient solution at outer iteration  $k$  is computed according to

$$\gamma = \begin{cases} \frac{1}{10}\epsilon & \text{if } k \bmod m = 0 \\ \frac{1}{10} \max(\epsilon, \min(1, \epsilon / \min(\|r^k\|_2, 1))) & \text{otherwise} \end{cases}$$

where  $\epsilon$  is the tolerance for the outer iteration. We have found the factor of 1/10 to be conservative in that this choice did not change the number of outer iterations needed to converge to  $\epsilon$  with the inner iteration tolerance fixed to  $\epsilon/100$  on the largest, most slowly converging problems. It may even be possible to relax this factor in some cases. Combining this approach with the flexible version of GMRES( $m$ ), FGMRES( $m$ ), enables the overall solution to be computed in the lowest possible number of iterations and with the least number of inner iterations, at the cost of a bit more storage. We also found that we could minimize the storage requirements by choosing the fairly low restart parameter of  $m = 10$  without significantly affecting the outer convergence, if at all.

### 3 NUMERICAL RESULTS

In this section we present the results of actual computations using the AttilaV2 DFEM first order transport code. We will compare the Krylov iterations to source (Richardson) iterations, with and without preconditioning using the partially consistent S-WLA DSA method. The first set of results are intended to show that the Krylov method actually does restore the effectiveness of the partially consistent DSA scheme, relative to source iteration. The second set of results are for a two-material “duct” problem to illustrate both how computational effort depends on scattering ratio and the total cross sections in the two regions in the problem.

We will not consider the FCDSA method described in Ref. 2 and analyzed in Part I because the expense associated with solving the consistently discretized  $P_1$  equations renders the method impractical, especially when compared to the partially consistent S-WLA method. We will refer to traditional source iteration that is accelerated by the S-WLA DSA method simply as “ASI”. The Krylov method approach will simply be called “PGMRES”, which is FGMRES(10) preconditioned with the S-WLA algorithm. If acceleration or a preconditioner are *not* used to accelerate the convergence of these iterations, respectively, then they will be referred to as “SI” or “GMRES”.

The following stopping criteria is used in all the results presented here:

$$\frac{\|\phi^{\ell+1} - \phi^\ell\|_2}{\|\phi^{\ell+1}\|_2} \leq 10^{-5}, \quad (20)$$

or 2000 iterations. Convergence criteria for the diagonally-scaled PCG inner iterations of the S-WLA algorithm is varied according to the size of the relative outer residual, as outlined in the previous section, for the Krylov iteration solutions. It is fixed to  $10^{-6}$  for the source iteration solutions. Initial scalar fluxes are zero.

The first set of computations use same mesh illustrated in Fig. 1. consisting of a  $(10 \times 10 \times 10)$  grid of cubes, 1.0 cm on a side, each divided into six tetrahedra of equal volume. The mesh contains two materials with total cross sections  $\sigma_{t,1}$  and  $\sigma_{t,2}$  in the two halves of every cube (each half being comprised of three

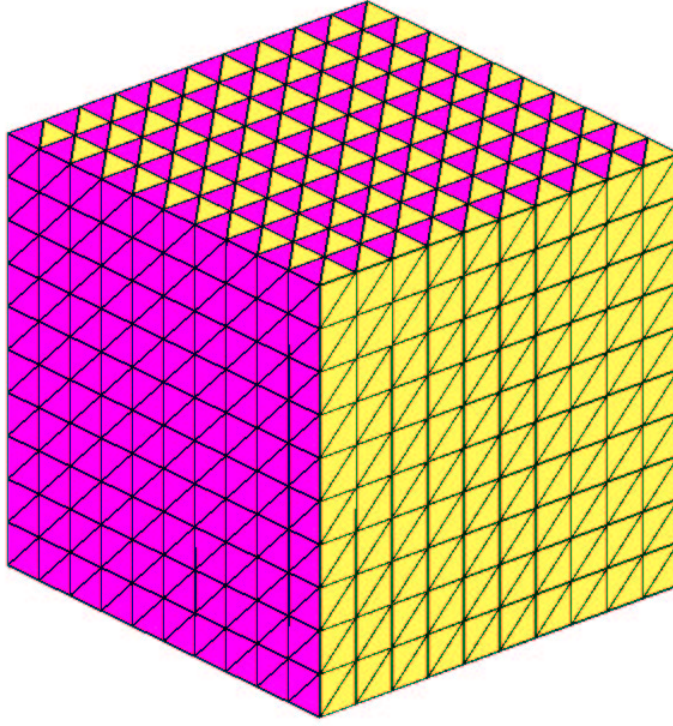


Figure 1: Two material mesh (indicated by the shading) consisting of a  $(10 \times 10 \times 10)$  grid of cubes, 1.0 cm on a side, divided into six tetrahedra each.

tetrahedra). This is similar to the mesh used in Part I (Fig. 4, Part I) for comparing the spectral radii from Fourier analysis to those measured with AttilaV2. Vacuum boundary conditions are applied on all six boundary faces. We computed the solution for this problem with a scattering ratio of  $c = 0.9999$  and an isotropic source of unit strength distributed throughout the problem. The number of iterations for  $\sigma_{t,1} = 2^{-10}, 2^0, 2^{10} \text{ cm}^{-1}$  with  $\sigma_{t,2} = 2^{-10}, \dots, 2^{10} \text{ cm}^{-1}$  is shown in Fig. 2. For this high scattering ratio, the Fourier analysis for the S-WLA scheme in Part I (Fig. 2(d), Part I) showed that the spectral radius approaches  $c$  as the difference in cross sections becomes large and This is reflected in the large number of source iterations in Fig. 2. It is clear that PGMRES restores much of the effectiveness of the partially consistent DSA method for the most slowly converging problems. Remarkably, PGMRES showed only a small variation in the number of iterations over this wide range of  $\sigma_{t,2}$ , needing anywhere from two to more than ten times fewer iterations than ASI to converge. Of course, this improvement comes at the extra cost associated with the FGMRES(10) iterations.

We will now investigate whether the reduction in the iteration counts is worth the additional computational effort of the Krylov method. This aim will best be served by considering a realistic problem. In this case, we will consider a duct problem consisting of a reflected quarter-cylinder, 25 cm in radius and 50 cm long. The “thin” duct region is 5 cm in radius. It is surrounded by a “thick” region and bends around a central disc of thick material. There is a unit isotropic boundary source incident on the left face of the duct.

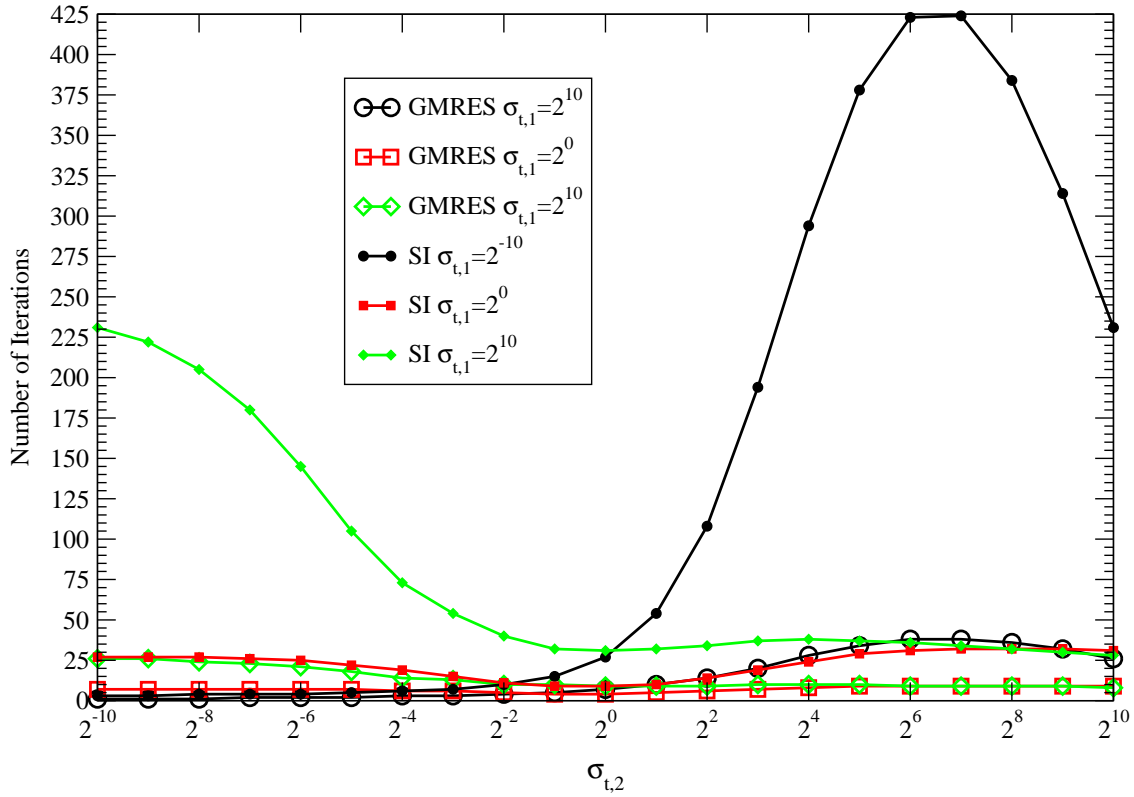


Figure 2: Iteration counts for FGMRES(10) (“GMRES”) and source iteration (“SI”) with the S–WLA method for  $c = 0.9999$ .

Vacuum boundary conditions are specified the outer surfaces. The values of the total cross sections in the thin region,  $\sigma_{t,1}$ , and the thick region,  $\sigma_{t,2}$ , as well as the scattering ratio  $c$ , will be varied to examine the effect of material heterogeneities in a realistic problem. An isotropic source of strength  $1.0^{-6}$  particles/cm<sup>3</sup> is distributed throughout the problem to help smooth the solution. The geometric configuration for two different meshes are illustrated in Figs. 3 and 4. One mesh consists entirely of 31,481 well-shaped tetrahedral cells, which we call the “Tet Mesh”. It is shown in Fig. 3. The other mesh has a layer of tetrahedra formed from dividing prisms that have been extruded from the thin region boundary into the thick region. We call this the “Prism Mesh”. It contains 118,211 cells and is shown in Fig. 4.

The number of iterations and the measured number of floating point operations (FLOP) on a single SGI Origin 2000 250 MHz CPU are tabulated below. The results for ASI and PGMRES on the Tet Mesh are shown in Tables 1 and 2. These results may be compared to the results for SI and GMRES (no DSA acceleration or preconditioning) on the Tet Mesh shown in Tables 3 and 4. The results for ASI and PGMRES on the Prism Mesh are shown Tables 5 and 6. Each table contains data for both solution methods for various values of thin ( $\sigma_{t,1}$ ) and thick ( $\sigma_{t,2}$ ) region cross sections and scattering ratios  $c = 1.0000, 0.9999, 0.999, 0.99, 0.9$ .

The most apparent and important observation is that PGMRES significantly reduces the number of iterations needed for convergence compared to ASI. This improvement in iteration count offsets any extra

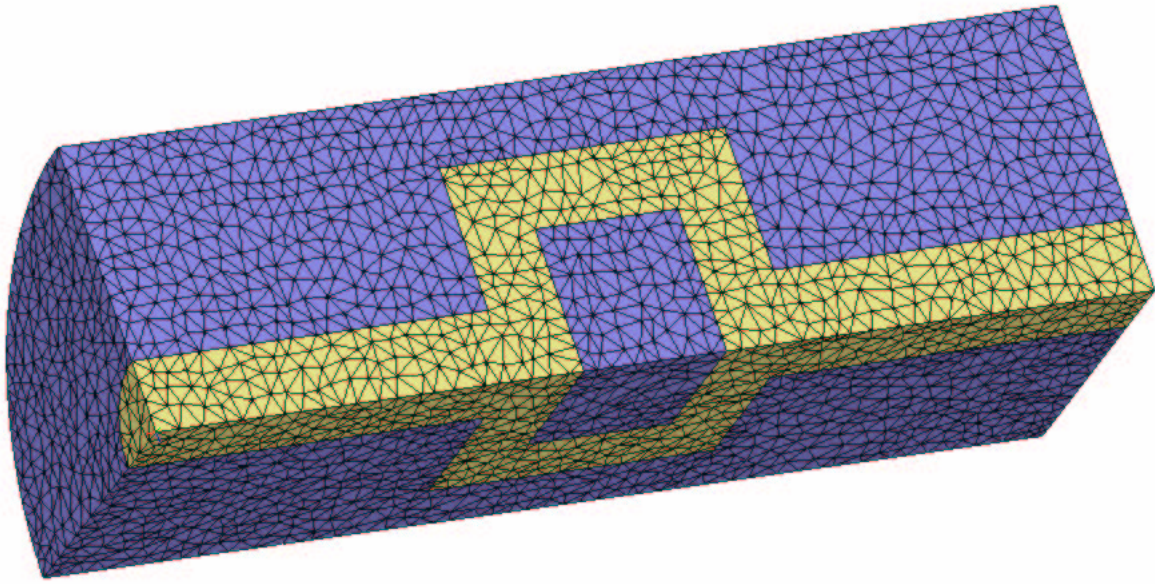


Figure 3: The Tet Mesh problem.

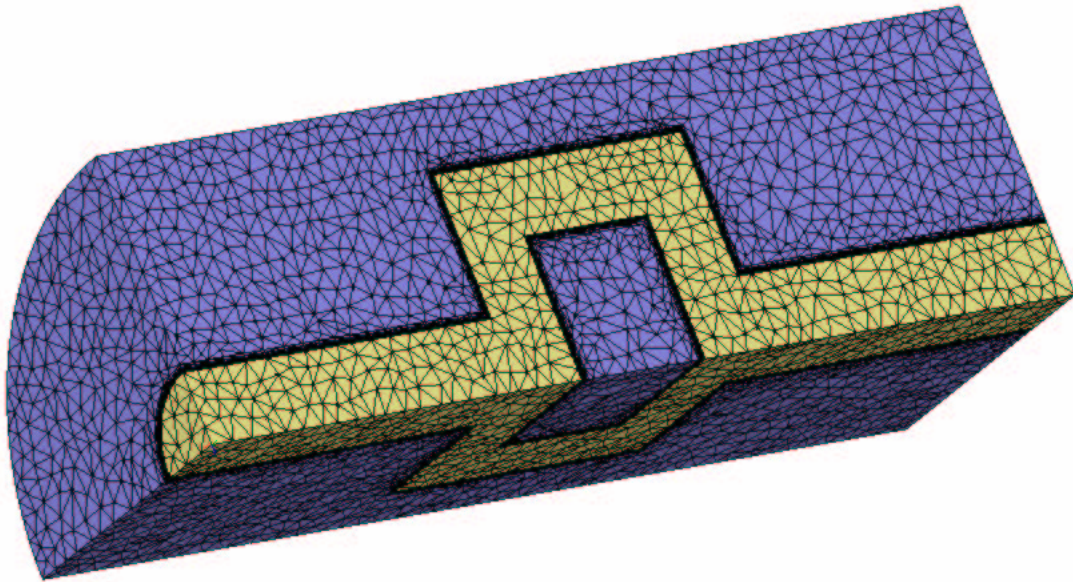


Figure 4: The Prism mesh problem. Note the layers of tetrahedra surrounding the duct formed from prisms extruded into the surrounding thick material.

computational effort associated with FGMRES(10), in all the cases shown here. The savings in computational effort compares more favorably as the scattering ratio  $c$  approaches 1.0 and S-WLA becomes less effective. While PGMRES needed less computational effort even for the lowest value of  $c = 0.9$  considered here, there is most likely going to be a point at which ASI, because it costs less per iteration than FGMRES(10), will compute a solution with less effort. However, as  $c$  approaches zero DSA isn't really needed anyway. It is notable that the results that FGMRES(10) alone, without the S-WLA preconditioner, does in fact accelerate the transport iterations. In some cases it even performs as well as DSA-accelerated source iteration, although we cannot conclude that this will be true in general. The results reported here are encouraging, especially since PGMRES outperforms source iteration even in homogeneous problems with equal total cross sections.

Finally, as shown in Table 7 the additional memory required for FGMRES(10) is not unreasonable, only about 50% greater than what is needed for the source iteration implementation.

## 4 CONCLUSIONS

The most remarkable observation from our numerical experiments is that the Krylov iterative method significantly improves efficiency of  $S_N$  transport calculations in problems for which a partially consistent but efficiently-computed DSA method is otherwise ineffective. Convergence significantly improves when FGMRES is preconditioned with the S-WLA DSA method, relative to either source iteration that is accelerated with S-WLA, or FGMRES without preconditioning.

The one-dimensional results reported in Refs. 3 and 4 both showed that preconditioned GMRES is very effective in reducing the number of iterations compared to DSA accelerated source iteration. The computational measurements in Ref. 4 indicated that the extra overhead associated with the Krylov method may not warrant its use. Our results on three-dimensional, unstructured tetrahedral meshes agree with the conclusion that preconditioned GMRES effectively reduces the iteration count relative to DSA accelerated source iteration and that GMRES alone, without preconditioning, improves convergence relative to unaccelerated source iterations. In contrast to these results, however, we found that any extra computational cost is outweighed by the reduction in the number of iterations. As  $c$  approaches zero, this relative advantage decreases and traditional source iteration probably becomes more efficient. The additional storage requirements for GMRES appear to be tolerable, a fact which should not change in time or energy dependent problems. Our results also indicate that GMRES alone, without DSA, improves convergence relative to unaccelerated source iteration for slowly converging problems. This confirms the suggestion in Refs.<sup>3</sup> and 15 that GMRES or some other Krylov method might be used to accelerate difficult problems in cases where the discretization or other considerations prohibit the use of DSA. Most importantly, however, our work shows that a partially consistent DSA method like the S-WLA method, serves as an excellent preconditioner for GMRES. This is true even in situations where the partially consistent scheme is no longer effective as an acceleration scheme

$\sigma_{t,1}$	$c$	$\sigma_{t,2}$							
		$10^3$		$10^2$		$10^1$		$10^0$	
		PGMRES	ASI	PGMRES	ASI	PGMRES	ASI	PGMRES	ASI
$10^{-3}$	1.0000	45	618	36	403	14	71	5	14
	0.9999	43	513	34	378	14	71	5	14
	0.999	25	222	30	291	14	69	5	14
	0.99	10	41	19	136	13	56	5	14
	0.9	4	9	8	28	8	28	5	12
$10^{-2}$	1.0000	36	379	29	267	13	59	5	13
	0.9999	31	305	28	251	13	58	5	13
	0.999	19	121	24	193	13	57	5	12
	0.99	9	34	16	85	12	46	5	12
	0.9	4	8	7	23	8	23	5	11
$10^{-1}$	1.0000	19	112	17	86	9	27	4	9
	0.9999	19	106	17	81	9	27	4	9
	0.999	17	83	15	67	9	26	4	9
	0.99	9	29	13	53	8	23	4	9
	0.9	4	8	7	19	6	17	4	8
$10^0$	1.0000	14	58	11	37	6	14	4	8
	0.9999	15	66	11	38	6	14	4	8
	0.999	14	59	12	39	6	14	4	8
	0.99	9	25	10	34	6	14	4	8
	0.9	5	10	6	16	5	12	4	7
$10^1$	1.0000	11	30	9	26	6	13		
	0.9999	10	36	9	27	6	13		
	0.999	10	37	9	30	6	13		
	0.99	8	24	9	27	6	13		
	0.9	6	15	7	16	6	12		
$10^2$	1.0000	11	35	11	34				
	0.9999	13	43	11	34				
	0.999	13	48	11	34				
	0.99	11	37	10	31				
	0.9	7	18	7	18				
$10^3$	1.0000	21	49						
	0.9999	13	50						
	0.999	12	44						
	0.99	9	28						
	0.9	4	10						

Table 1: Computational results for the Tet Mesh. Number of iterations are tabulated for total cross sections  $\sigma_{t,1}$  and  $\sigma_{t,2}$  ( $\text{cm}^{-1}$ ) and a range of scattering ratio  $c$ .

$\sigma_{t,1}$	$c$	$\sigma_{t,2}$							
		$10^3$		$10^2$		$10^1$		$10^0$	
		PGMRES	ASI	PGMRES	ASI	PGMRES	ASI	PGMRES	ASI
$10^{-3}$	1.0000	8.17	90.05	6.54	58.79	2.94	10.43	1.30	2.13
	0.9999	7.86	74.31	6.24	55.11	2.94	10.43	1.30	2.13
	0.999	4.75	32.06	5.48	42.18	2.94	10.15	1.30	2.13
	0.99	2.14	5.92	3.60	19.66	2.75	8.23	1.30	2.13
	0.9	1.18	1.35	1.74	4.07	1.84	4.12	1.29	1.84
$10^{-2}$	1.0000	6.46	54.70	5.27	38.51	2.66	8.57	1.28	1.96
	0.9999	5.69	43.79	5.11	36.12	2.66	8.43	1.28	1.96
	0.999	3.57	17.33	4.47	27.71	2.65	8.28	1.28	1.81
	0.99	1.87	4.87	3.09	12.19	2.51	6.68	1.27	1.81
	0.9	1.05	1.17	1.56	3.32	1.73	3.35	1.27	1.66
$10^{-1}$	1.0000	3.62	16.01	3.23	12.29	1.95	3.89	1.07	1.32
	0.9999	3.61	15.10	3.22	11.58	1.95	3.89	1.07	1.32
	0.999	3.27	11.73	2.90	9.56	1.95	3.75	1.07	1.32
	0.99	1.86	3.96	2.59	7.51	1.79	3.32	1.07	1.32
	0.9	1.05	1.06	1.48	2.63	1.46	2.44	1.07	1.18
$10^0$	1.0000	2.70	8.15	2.25	5.19	1.34	1.96	1.02	1.12
	0.9999	2.79	9.11	2.26	5.31	1.34	1.96	1.02	1.12
	0.999	2.59	7.82	2.37	5.35	1.33	1.96	1.02	1.12
	0.99	1.64	3.09	1.89	4.54	1.32	1.94	1.02	1.12
	0.9	1.06	1.25	1.23	2.00	1.14	1.61	1.01	0.99
$10^1$	1.0000	2.54	4.11	1.72	3.50	1.35	1.73		
	0.9999	1.93	4.70	1.70	3.60	1.35	1.73		
	0.999	1.85	4.51	1.67	3.87	1.34	1.72		
	0.99	1.53	2.87	1.60	3.28	1.31	1.67		
	0.9	1.25	1.79	1.30	1.91	1.26	1.46		
$10^2$	1.0000	2.24	4.65	2.04	4.34				
	0.9999	2.21	5.14	1.99	4.17				
	0.999	2.17	5.55	1.94	3.97				
	0.99	1.90	4.20	1.68	3.54				
	0.9	1.27	2.07	1.27	2.07				
$10^3$	1.0000	4.48	6.07						
	0.9999	2.22	5.65						
	0.999	2.09	4.94						
	0.99	1.61	3.17						
	0.9	0.98	1.18						

Table 2: Computational results for the Tet Mesh. FLOP counts (in billions) are tabulated for total cross sections  $\sigma_{t,1}$  and  $\sigma_{t,2}$  ( $\text{cm}^{-1}$ ) and a range of scattering ratio  $c$ .

$\sigma_{t,1}$	$c$	$\sigma_{t,2}$							
		$10^3$		$10^2$		$10^1$		$10^0$	
		GMRES	SI	GMRES	SI	GMRES	SI	GMRES	SI
$10^{-3}$	1.0000	112	n/c	350	n/c	43	606	8	28
	0.9999	88	n/c	316	n/c	42	601	8	28
	0.999	33	n/c	85	n/c	41	553	8	28
	0.99	12	486	30	441	28	310	8	27
	0.9	4	59	10	67	12	60	7	22
$10^{-2}$	1.0000	325	n/c	319	n/c	46	607	8	28
	0.9999	108	n/c	213	n/c	46	601	8	28
	0.999	42	n/c	92	n/c	44	553	8	28
	0.99	13	486	33	441	29	310	8	27
	0.9	5	59	10	67	13	60	7	22
$10^{-1}$	1.0000	156	n/c	322	n/c	45	609	8	28
	0.9999	93	n/c	269	n/c	44	603	8	28
	0.999	50	n/c	116	n/c	42	555	8	28
	0.99	18	485	35	441	29	311	8	27
	0.9	6	59	12	67	13	60	7	22
$10^0$	1.0000	1133	n/c	393	n/c	50	635	10	35
	0.9999	320	n/c	291	n/c	50	628	10	35
	0.999	73	n/c	128	n/c	47	575	10	35
	0.99	26	477	39	441	33	313	9	34
	0.9	9	59	14	66	14	60	8	25
$10^1$	1.0000	920	n/c	660	n/c	57	850		
	0.9999	401	n/c	392	n/c	57	835		
	0.999	99	n/c	170	n/c	52	720		
	0.99	37	439	45	439	33	319		
	0.9	14	57	16	65	14	61		
$10^2$	1.0000	n/c	n/c	769	n/c				
	0.9999	549	n/c	363	n/c				
	0.999	136	n/c	131	n/c				
	0.99	37	441	37	452				
	0.9	12	68	11	68				
$10^3$	1.0000	146	n/c						
	0.9999	49	n/c						
	0.999	25	1776						
	0.99	11	511						
	0.9	5	81						

Table 3: Computational results for the Tet Mesh *without* S-WLA preconditioning or acceleration. Number of iterations are tabulated for total cross sections  $\sigma_{t,1}$  and  $\sigma_{t,2}$  ( $\text{cm}^{-1}$ ) and a range of scattering ratio  $c$ . An entry “n/c” indicates that the problem did not converge in 2000 iterations.

$\sigma_{t,1}$	$c$	$\sigma_{t,2}$							
		$10^3$		$10^2$		$10^1$		$10^0$	
		GMRES	SI	GMRES	SI	GMRES	SI	GMRES	SI
$10^{-3}$	1.0000	14.73	n/c	44.43	n/c	5.94	63.25	1.32	3.00
	0.9999	11.60	n/c	40.14	n/c	5.83	62.72	1.32	3.00
	0.999	4.65	n/c	11.04	n/c	5.71	57.72	1.32	3.00
	0.99	1.98	50.74	4.09	46.05	3.98	32.39	1.32	2.90
	0.9	0.93	6.23	1.57	7.07	1.99	6.34	1.20	2.38
$10^{-2}$	1.0000	41.20	n/c	40.33	n/c	6.15	63.35	1.32	3.00
	0.9999	13.90	n/c	27.09	n/c	6.15	62.72	1.32	3.00
	0.999	5.66	n/c	11.93	n/c	5.91	57.72	1.32	3.00
	0.99	1.99	50.74	4.50	46.05	3.98	32.39	1.32	2.90
	0.9	0.96	6.23	1.56	7.07	2.01	6.34	1.20	2.38
$10^{-1}$	1.0000	20.21	n/c	40.75	n/c	6.18	63.56	1.32	3.00
	0.9999	12.27	n/c	34.04	n/c	6.06	62.93	1.32	3.00
	0.999	6.77	n/c	14.87	n/c	5.83	57.93	1.32	3.00
	0.99	2.69	50.63	4.72	46.05	4.10	32.50	1.32	2.90
	0.9	1.17	6.23	1.89	7.07	2.11	6.34	1.20	2.38
$10^0$	1.0000	142.68	n/c	49.88	n/c	6.63	66.27	1.57	3.73
	0.9999	40.51	n/c	37.05	n/c	6.63	65.54	1.57	3.73
	0.999	9.54	n/c	16.41	n/c	6.27	60.01	1.57	3.73
	0.99	3.59	49.80	5.22	46.05	4.54	32.71	1.45	3.63
	0.9	1.43	6.23	2.11	6.96	2.12	6.34	1.32	2.69
$10^1$	1.0000	118.08	n/c	83.66	n/c	7.70	88.68		
	0.9999	51.70	n/c	49.88	n/c	7.70	87.11		
	0.999	13.05	n/c	21.78	n/c	7.11	75.13		
	0.99	5.13	45.84	6.03	45.84	4.66	33.33		
	0.9	2.22	6.02	2.35	6.86	2.22	6.44		
$10^2$	1.0000	252.58	n/c	97.31	n/c				
	0.9999	69.78	n/c	46.15	n/c				
	0.999	17.47	n/c	16.93	n/c				
	0.99	4.99	46.05	5.00	47.19				
	0.9	1.89	7.17	1.78	7.17				
$10^3$	1.0000	19.14	n/c						
	0.9999	6.68	n/c						
	0.999	3.61	185.19						
	0.99	1.87	53.34						
	0.9	1.05	8.53						

Table 4: Computational results for the Tet Mesh *without* S-WLA preconditioning or acceleration. FLOP counts (in billions) are tabulated for total cross sections  $\sigma_{t,1}$  and  $\sigma_{t,2}$  ( $\text{cm}^{-1}$ ) and a range of scattering ratio  $c$ . An entry “n/c” indicates that the problem did not converge in 2000 iterations.

$\sigma_{t,1}$	$c$	$\sigma_{t,2}$							
		$10^3$		$10^2$		$10^1$		$10^0$	
		PGMRES	ASI	PGMRES	ASI	PGMRES	ASI	PGMRES	ASI
$10^{-3}$	1.0000	21	123	17	121	12	53	5	14
	0.9999	25	131	17	113	12	53	5	14
	0.999	24	110	15	88	12	52	5	14
	0.99	18	80	12	53	11	42	5	14
	0.9	9	28	8	23	7	22	5	12
$10^{-2}$	1.0000	21	108	17	107	11	46	5	13
	0.9999	25	119	16	99	11	46	5	13
	0.999	24	107	14	77	11	45	5	13
	0.99	19	80	12	47	10	37	5	13
	0.9	9	28	8	21	7	19	5	11
$10^{-1}$	1.0000	19	78	12	55	9	25	4	9
	0.9999	24	104	12	52	8	25	4	9
	0.999	23	103	12	43	8	24	4	9
	0.99	17	79	10	31	7	19	4	9
	0.9	9	28	8	19	6	13	4	8
$10^0$	1.0000	17	76	10	30	5	11	4	8
	0.9999	20	104	10	31	5	11	4	8
	0.999	19	102	10	32	5	11	4	8
	0.99	16	79	10	31	5	11	4	8
	0.9	8	28	8	20	5	10	4	7
$10^1$	1.0000	21	87	10	32	6	13		
	0.9999	21	114	10	33	6	13		
	0.999	21	115	10	35	6	13		
	0.99	17	88	10	35	6	13		
	0.9	9	31	8	24	6	11		
$10^2$	1.0000	22	111	13	42				
	0.9999	25	146	12	43				
	0.999	25	156	13	45				
	0.99	19	109	12	44				
	0.9	11	34	9	27				
$10^3$	1.0000	31	145						
	0.9999	31	182						
	0.999	29	173						
	0.99	21	109						
	0.9	11	36						

Table 5: Computational results for the Prism Mesh. Number of iterations are tabulated for total cross sections  $\sigma_{t,1}$  and  $\sigma_{t,2}$  ( $\text{cm}^{-1}$ ).

$\sigma_{t,1}$	$c$	$\sigma_{t,2}$							
		$10^3$		$10^2$		$10^1$		$10^0$	
		PGMRES	ASI	PGMRES	ASI	PGMRES	ASI	PGMRES	ASI
$10^{-3}$	1.0000	15.13	66.80	12.13	66.05	9.64	29.11	4.81	8.02
	0.9999	17.09	70.01	12.10	61.35	9.64	29.10	4.81	8.02
	0.999	16.46	58.32	10.84	47.35	9.63	28.53	4.81	8.02
	0.99	12.54	42.12	9.15	28.41	9.04	22.95	4.81	8.02
	0.9	7.11	14.82	6.30	12.37	6.13	12.04	4.79	6.91
$10^{-2}$	1.0000	14.64	57.81	11.97	57.66	8.78	25.30	5.14	8.07
	0.9999	16.64	62.86	11.39	53.15	8.77	25.30	5.14	8.07
	0.999	15.94	56.16	10.23	41.11	8.74	24.73	5.14	8.07
	0.99	12.65	41.69	9.05	24.99	7.59	20.32	5.14	8.07
	0.9	6.71	14.66	6.22	11.22	5.81	10.47	5.11	6.86
$10^{-1}$	1.0000	13.10	41.34	9.18	29.73	7.76	15.05	5.67	7.21
	0.9999	16.15	54.37	9.16	28.09	7.19	15.05	5.67	7.21
	0.999	15.47	53.41	9.08	23.12	7.20	14.46	5.66	7.21
	0.99	11.60	40.53	7.37	16.54	6.53	11.40	5.64	7.21
	0.9	6.83	14.31	6.11	10.05	5.81	7.72	5.60	6.41
$10^0$	1.0000	11.29	39.88	7.79	17.36	6.19	8.48	6.61	7.78
	0.9999	12.80	53.61	7.73	17.82	6.19	8.47	6.61	7.78
	0.999	11.98	51.48	7.65	18.04	6.19	8.44	6.62	7.79
	0.99	10.26	38.70	7.34	16.88	6.03	8.26	6.60	7.77
	0.9	5.61	13.33	5.88	10.29	5.43	6.80	6.37	6.71
$10^1$	1.0000	15.97	48.13	8.85	22.83	7.76	11.43		
	0.9999	14.43	59.12	8.55	23.05	7.76	11.40		
	0.999	13.28	55.44	7.95	21.88	7.62	11.20		
	0.99	10.33	39.37	7.09	18.50	6.85	9.76		
	0.9	5.98	13.39	5.53	11.33	5.56	6.38		
$10^2$	1.0000	16.10	75.38	12.41	35.22				
	0.9999	15.11	72.63	10.65	30.36				
	0.999	14.29	70.71	9.58	25.18				
	0.99	10.73	47.04	8.25	21.59				
	0.9	6.93	14.46	5.88	12.35				
$10^3$	1.0000	27.74	103.88						
	0.9999	19.12	84.31						
	0.999	17.12	76.50						
	0.99	12.23	46.46						
	0.9	7.19	15.15						

Table 6: Computational results for the Prism Mesh. FLOP counts (in billions) are tabulated for total cross sections  $\sigma_{t,1}$  and  $\sigma_{t,2}$  ( $\text{cm}^{-1}$ ).

Mesh	PGMRES	GMRES	ASI	SI
Tet	276	246	181	178
Prism	925	840	639	632

Table 7: Approximate memory requirements (MB) measured on dedicated SGI Origin 2000 single processors for the two duct problem meshes.

for traditional source iteration.

It is worth noting again that the Krylov method approach can easily be implemented in transport codes using the existing  $S_N$  space and angle sweeps and other machinery already present in a given code. This is true at least for multi-group energy dependence with downscatter. Although the results reported here considered only isotropic scattering, the method is implemented for general scattering anisotropy. Higher orders of scattering are easily accounted for by simply increasing the size of the Krylov vectors to include the additional flux moments needed to compute the scattering source. The rest of the original code remaining unchanged. It is also possible that outer iterations can be replaced by a Krylov iteration but this would require more extensive modification. These considerations will be examined in the future.

There are many questions that remain. What is it about the eigenvalue (or singular value) distribution of the preconditioned linear system corresponding to the transport operator that enables GMRES to improve the effectiveness of otherwise degraded DSA methods? Could our conclusions change or the optimality of both accelerated source iteration or the Krylov iterative method approach be improved if an (algebraic) multigrid method is used to solve the continuous diffusion equation of the partially consistent method? What is the effect on the accuracy of the converged Krylov method solution and how does this affect the solution in deep penetration problems, for example? Finally, we emphasize that we have shown that this method works very well for our particular discretization technique and DSA scheme on certain problems of interest and for unstructured tetrahedral meshes. It will be interesting to find out how it performs with other types of discretizations, DSA methods and meshes.

### Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy at the Los Alamos National Laboratory. The authors would like to thanks Marvin Adams of Texas A&M University and James Holloway and Ed Larsen of the University of Michigan for their very helpful discussions.

## References

1. T. A. Wareing, J. M. McGhee, J. E. Morel, and S. D. Pautz, “Discontinuous Finite Element  $S_n$  Methods on Three-Dimensional Unstructured Grids,” *Nucl. Sci. Engr.*, **138**, pp. 256–268 (2001).
2. J. S. Warsa, T. A. Wareing, and J. E. Morel, “Fully Consistent Diffusion Synthetic Acceleration of Linear Discontinuous Transport Discretizations on Three-Dimensional Unstructured Meshes,” *Nucl. Sci. Engr.*, **141**, pp. 236–251 (2002).
3. S. F. Ashby, P. N. Brown, M. R. Dorr, and A. C. Hindmarsh, “Preconditioned Iterative Methods for Discretized Transport Equations,” in **Proc. International Topical Meeting on Advances in Mathematics, Computations, Reactor Physics**, 28 April – 2 May, Vol. 2, Pittsburgh, Pennsylvania, pp. 6.1 2–1 (1991). (Also available as LLNL Report UCRL-JC-104901, July 1990.).
4. B. Guthrie, J. P. Holloway, and B. W. Patton, “GMRES as a Multi-Step Transport Sweep Accelerator,” *Tran. Theory and Stat. Phys.*, **28**, n. 1, pp. 83–102 (1999).
5. T. A. Wareing, “New Diffusion-Synthetic Accelerations Methods for the  $S_N$  Equations with Corner Balance Spatial Differencing,” in **Joint International Conference on Mathematical Methods and Supercomputing in Nuclear Applications**, 19-23 April, Vol. 2, Karlsruhe, Germany, p. 500 (1993).
6. V. Faber and T. A. Manteuffel, **A Look at Transport Theory from the Point of View of Linear Algebra**. in *Lecture Notes in Pure and Applied Mathematics* (P. Nelson, et al., Ed.), Vol. 115, pp. 31–61 Marcel Dekker: New York (1989).
7. C. T. Kelley, “Multilevel Source Iteration Accelerators for the Linear Transport Equation in Slab Geometry,” *Trans. Theory and Stat. Phys.*, **24**, n. 4&5, pp. 697–707 (1995).
8. C. T. Kelley and Z. Q. Xue, “GMRES and Integral Operators,” *SIAM J. Scientific Computing*, **17**, n. 1, pp. 217–226 (1996).
9. S. Oliveira and Y. Deng, “Preconditioned Krylov Subspace Methods for Transport Equations,” *Prog. Nucl. Energy*, **33**, n. 1/2, pp. 155–174 (1998).
10. Y. Saad, **Iterative Methods for Sparse Linear Systems**. PWS Publishing Company: Boston (1996).
11. B. W. Patton and J. P. Holloway, “Application of Krylov Subspace Methods to the Slab Geometry Transport Equation,” in **ANS Topical Meeting on Radiation Protection and Shielding**, 21-25 April, Vol. 1, Cape Cod, pp. 384–389 (1996).

12. R. Sanchez and S. Santandrea, “Symmetrization of the Transport Operator and Lanczos’ Iterations,” in **Proceedings of the 2001 International Meeting on Mathematical Methods for Nuclear Applications**, 9-13 Sep., Salt Lake City, Utah (2001).
13. B. W. Patton and J. P. Holloway, “Application of Preconditioned GMRES to the Numerical Solution of the Neutron Transport Equation,” *Ann. Nucl. Energy*, **29**, n. 2, pp. 109–136 (2002).
14. S. F. Ashby, P. N. Brown, M. R. Dorr, and A. C. Hindmarsh, “A Linear Algebraic Analysis of Diffusion Synthetic Acceleration for the Boltzmann Transport Equation,” *SIAM J. Num. Anal.*, **32**, pp. 128–178 (1995).
15. P. N. Brown, “A Linear Algebraic Development of Diffusion Synthetic Acceleration for Three-Dimensional Transport Equations,” *SIAM J. on Num. Anal.*, **32**, pp. 179–214 (1995).
16. A. Greenbaum, **Iterative Methods for Solving Linear Systems**. SIAM: Philadelphia (1997).
17. G. L. Ramone, M. L. Adams, and P. F. Nowak, “A Transport Synthetic Acceleration Method for Transport Iterations,” *Nucl. Sci. Engr.*, **125**, pp. 257–283 (1997).
18. M. R. Zika and M. L. Adams, “Transport Synthetic Acceleration with Opposing Reflecting Boundary Conditions,” *Nucl. Sci. Engr.*, **134**, pp. 159–170 (2000).
19. G. Meurant, **Computer Solution of Large Linear Systems**. in *Studies in Mathematics and Applications*, Vol. 28 Elsevier: Amsterdam (1999).
20. J. S. Warsa, M. Benzi, T. A. Wareing, and J. E. Morel, “Preconditioning a Mixed Discontinuous Finite Element Method for Radiation Diffusion,” *Num. Lin. Alg. Appl* (2001), to appear.
21. Y. Saad, “A Flexible Inner-Outer Preconditioned GMRES Algorithm,” *SIAM J. Sci. Stat. Comp.*, **14**, pp. 461–469 (1993).
22. Y. Notah, “Flexible Conjugate Gradients,” *SIAM J. Sci. Stat. Comp.*, **22**, pp. 1444–1460 (2000).
23. G. H. Golub and Q. Ye, “Inexact Preconditioned Conjugate Gradient Method with Inner-Outer Iteration,” *SIAM J. Sci. Comput.*, **21**, pp. 1305–1320 (1999).
24. A. Bouras and V. Frayssé, “A Relaxation Strategy for Inexact Matrix-Vector Products for Krylov Methods,” CERFACS TR/PA/00/15, European Centre for Research and Advanced Training in Scientific Computation, Toulouse, France (2000). Submitted to *SIAM J. Matrix Anal. Appl.*.